Predicting Local Congestion at Fine-grain Levels in Networks-on-Chip Using Spiking Neural Networks

Predicting Local Congestion at Fine-grain Levels in Networks-on-Chip Using Spiking Neural Networks

Aqib Javed*, Jim Harkin, Liam McDaid and Junxiu Liu
School of Computing, Engineering and Intelligent Systems,
Ulster University, Magee Campus, Derry, Northern Ireland, United Kingdom, UK
*Contact: Javed-a@ulster.ac.uk

***Abstract*—Networks-on-Chip (NoC) was introduced to achieve maximum communication performance in on-chip systems. Local congestion caused by the queuing of data at input channel buffers constrains NoC latency and throughput performance. NoCs require a predictive approach to minimize the effects from local congestion problems. In this paper we proposed a novel fine-grain congestion prediction approach based on Spiking Neural Network (SNN), which predicts router congestion with 30 clock cycles look-ahead capability. Two fine-grain prediction approaches including router and network models are proposed. The prediction performances of the models are evaluated and analyzed using both synthetic and real-time NoC traffic applications. Results show that the network model is more consistent in fine-grain local congestion prediction and requires 42% less hardware area than the router model.**

Keywords— Networks-on-Chip; congestion prediction; network traffic; Spiking Neural Networks

I. INTRODUCTION

Multi-Processor SoC (MPSoC) are designed to execute complex applications in parallel to achieve high computational performance. SoCs require communication architectures to transmit data between logical components [1], [2]. Traditionally, shared-bus architectures are used in SoC for data transmission. With an increase in the number of logical elements on chips, shared-bus based architectures cause latency issues which restricts SoC in achieving desired performance [3].

Networks-on-Chip (NoC) were introduced as a potential remedy for poor scalability and latency issues caused by shared-bus based systems. Quality of Service (QoS) is an important parameter to ensure interconnect performance [4]. NoC offers concurrent communication paths for data transmission to enhance QoS and ensures maximum network throughput [5]. It also provides a number of interconnection topologies to accommodate over one thousand cores [6]. NoCs are equipped with additional data traffic management resources, i.e. routing algorithm, network topology, flow control etc., to enhance communication performance. Ideally, the NoC is designed to distribute traffic uniformly across the communication network. The ability to balance the distribution of traffic across a NoC is constrained due to non-optimized application mapping across network cores and the selection of inappropriate traffic management resources [7]. This non-uniform traffic distribution pushes data packets to flow towards specific nodes which causes communication delays and ultimately leads to congestion. Congestion is an important factor in NoC performance degradation and needs to be addressed at an early stage to minimize its impact on NoC throughput [8].

Neural Networks (NNs) are mathematical counterparts of biological neurons. Since their evolution, NNs have shown an excellent performance in the field of data learning and classification [9], [10]. Biological neurons generate action potential (spikes) to transmit information towards connected neuron through dendrites (synapses). These biological neurons are connected together in the form of a complex neural network to encode (weighted) information, where task of classification and prediction has been demonstrated. Spiking Neural Networks (SNNs) are inspired from biological neurons to encode highly complex tasks in a spatial domain [1], [11], [12]. Contrary to Artificial Neural Network (ANNs), SNNs are multilayer network of connected spiking neurons to encode information in the form of synapses (temporal manner). Recent studies show that SNNs are computationally more powerful and hardware efficient when compared to ANNs[13]–[15]. NoC is a digital system that generates temporal communication patterns and SNN can use these temporal information to predict potential congestion hotspot in NoC [9]. In addition, a key motivation for using SNNs is the significantly reduced area overhead compared with ANNs [15]. This is a key scalability criterion for modern systems.

NoC communication performance can be improved by predicting local congestion prior to its occurrence [8], [16], [17]. Existing NoC congestion prediction models classify on-path network node as congested/non-congested based on its potential congestion status [13], [16], [18]. This bi-level congestion output is forwarded to adaptive routing algorithms/congestion handlers to make routing decisions. Problem arise when router is facing multiple congestion paths and routing algorithm is unable to analyses the depth of on-path node congestion in order to make routing decision. This work proposes a novel SNN based multi-level, fine-grain prediction strategy to predict fine-grain buffer utilization for each routing node within 30 clocks cycles in advance of any potential congestion. The predicted utilization output will aid congestion handling mechanism (adaptive routing) in finding an optimal routing path (minimal latency) under potential congestion conditions. This work proposed two SNN based prediction models: router-model and network-model to explore low-cost and high performance congestion prediction mechanism for NoCs. Models are evaluated in terms of fine-grain prediction accuracy on synthetic and real-time traffic applications. The primary objective of this work is to provide an optimal multi-

level fine-grain congestion prediction model that will predict level of local congestion to enhance NoC performance.

Section II provides background on NoC congestion and existing congestion detection\prediction models. Section III presents the proposed prediction model. An experimental setup to analyse performance of proposed models are explained in section IV. Section V presents simulation results. Section VI provides a conclusion and outlines future work.

II. BACKGROUND AND LITERATURE REVIEW

In MPSoC networks, NoC latency and QoS depends on number of parameters: routing algorithms, application mapped, network topology etc. These parameters influence on-chip routing behavior that causes network congestion [8]. NoC congestion is not an instant phenomenon, it evolves in phases before it spreads throughout network. NoC congestion starts from switch contention inside routers and then spreads outside towards the neighboring nodes [19]. Switch contention queues incoming data packets at respective input channel buffers. Routers are equipped with limited buffers are soon run out of buffering space thus requesting neighboring nodes to stop sending data towards it. This queuing of data packets cause backpressure effect towards the neighboring nodes. Thus, a contention caused inside a router triggers global NoC congestion. Fig 1 shows the backpressure effect in the NoCs. Insertion of additional buffering spaces at input channels can minimize the effect of backpressure but cause huge communication delays. The likelihood is that backpressure still persists if these buffers are filled [19]. Therefore, the NoC requires a concrete solution for unbalance traffic load to avoid the creation of congestion.

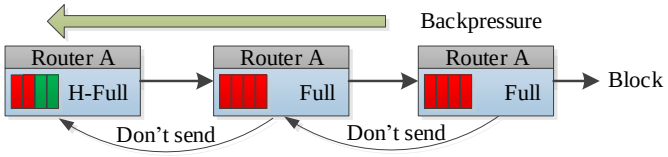


Fig.1. Effect of congestion and Backpressure.

NoC data traffic can be handled by adopting the appropriate data flow mechanisms to optimize traffic flow [20]. NoC is equipped with a Worm-hole Flow Control (WFC) mechanism to packetize data traffic into small chunks (flits). Routing channels are established as soon as the header flit is received by the next in-path router. This channel allocation queue other data packets at respective input buffers that adds into communication delays and congestion issues. Virtual channels (VC) can be employed as a remedy for delays caused by WFC [21]. Recent research highlights the devastating effects of local congestion on NoC performance [8]. Data encountered with in-path congestion significantly effects network latency. Local congestion handled at early stage will improves overall NoC performance [8].

In NoC systems, applications are mapped across all network nodes and data packets generated by each node is dependent on the routing algorithm to reach destination node. Routing algorithms are responsible for uniform distribution of data across network. Different routing algorithms i.e. XY, Odd-Even etc. are designed for the NoC architecture [22]. These routing algorithms are static-natured and lacks capability to bypass an emerging congestion hazard. An adaptive routing algorithms are introduced with an ability to select the next hop towards destination based on current network condition. Dynamic XY (DyXY), Dynamic AD (DyAD) etc. are widely used adaptive routing algorithms [23], [24]. These routing algorithms requires local information to guarantee routing data transmission through minimal path. These algorithms incurs extra logical circuit costs to decide for the optimal path. Furthermore, these routing algorithms comes with the local visibility that lacks information of far-neighboring nodes and often forward data packets towards nodes which are already congested (misjudgment problem) or traverse data through additional hops thus effecting NoC latency [7], [25].

Dedicated Congestion Aware Adopting Routing (CAAR) algorithms are designed to tackle misjudgment and latency issues caused by simplified adaptive routing algorithms. CAAR algorithms are equipped with complex selection functions to analyze the spread of NoC congestion. These selection function use local or global information to optimize data path [26]. Some CAAR algorithms considers switch contention as a potential congestion identifier. An upgraded Odd-Even adaptive routing algorithm and Path-Congestion Aware Adaptive Routing (PCAR) algorithm used switch contention and free buffer slots information as a selection function to identify on-path congestion [19]. Other prominent solution includes dynamic run-time task mapping to enhance network throughput[27]. These complex algorithms improve network latency with cost of additional computational and hardware resources.

All of above techniques are reactive to congestion and enforce precautionary measurements to avoid data packets from existing congestion. NoC performance i.e. latency, throughput, QoS can be improved by predicting on-path congestion[8], [13], [16]. NoC congestion prediction is an active research topic and recent studies reveal that congestion prediction can aid routing algorithms in making optimal routing decisions to bypass congested nodes. This precautionary-routing will reduce the impact of congestion on NoC performance by minimizing the possibility of potential local congestion.

A. Congestion Prediction Models

Congestion prediction is an on-going research topic and previous work utilized different indicators as a parameters to predict congestion in NoCs. These parameters includes buffer occupancy levels, traffic patterns, traffic tables, task mapping etc. Traffic-Based Routing Algorithm (TBRA) analyses routing data to predicts on-path congestion and routes the data packets through alternative paths by dynamically select suitable routing algorithms [7]. Advantages of congestion prediction are not limited to linearize network traffic distribution to optimize network performance (latency and throughput), it also helps in optimization of NoC resource utilization thus saving dynamic power. Application Driven Traffic Pattern Table (ATPT) with build-in traffic flow table predicts end-to-end network traffic patterns. ATPTs process incoming traffic flow patterns to predict router usage. Based on predicted router usage, ATPT optimizes the operating frequency/voltage of router to save

86% dynamic power with cost of 21% NoC latency[28]. A predictive flow control mechanisms is proposed to ensure congestion free paths for routing data. This close-loop flow control mechanism avoids buffer overflowing by dropping random data packets at each router[29].

Neural networks have achieved excellent performance in predicting NoC congestion. A multi layered Evolving Fuzzy Neural Network (EFuNN) predicts on-path congestion to provide congestion-free minimal paths [17]. An ANN based hotspot prediction mechanism was proposed for mesh based NoC system that utilizes buffer occupancy levels as a parameter for congestion prediction. The model achieved 65-92% congestion prediction accuracy on real-time and synthetic traffic patterns. Recent research on analyzing SNN based congestion prediction models has achieved up to 88%-96.59% prediction accuracy on synthetic and real-time applications, with 9 times less hardware overhead as compared to ANN [13].

B. Motivation:

Existing congestion prediction models classify router accessibility based on congestion and non-congestion status. Typically, a threshold is set on prediction parameter to classify router as congested or non-congested. Predicted congested/non-congested output are then forwarded to routing algorithms to route data through alternative paths.

Consider a situation shown in Fig. 2 where the data packet is forwarded from R11 to R12 while coursed towards R23. Depending on the routing definition, data arrived at R12 can be forwarded through R22, R13 and R02 towards destination node (as shown in Fig. 2(a)). Adaptive routing algorithms have congestion criteria/definition to look at on-path congestion and to make optimal decision for routing of the data traffic towards least congested node. CAAR and existing congestion prediction models sets the threshold condition on input buffer levels to define congestion and predict router status as congested or non-congested. These predictive outputs are then fed into the routing algorithm to make precautionary measurements i.e. establish an alternative path for data to avoid potential congestion. According to congestion definition in [13], [16], all neighboring routers (R22, R13 and R23) are congested and routing algorithms have no visibility to the actual router utilization values (as shown in Fig. 2(b)). Thus, the data packet received by R12 will be queued at its input buffer until one of the on-path nodes gets freed from congested status. Queuing of incoming data will soon occupy all available buffering space and starts causing backpressure on neighboring nodes. Other possible solutions by the routing algorithm is the reverting of data packet towards R11. Thus, allowing R11 to find alternative non-congested path for routing data packet. This process will add additional hops in data transmission which will cause communications delays and NoC latency issues.

This sparks the motivation of our research to evaluate prediction performance for multi-level fine-grain utilization level that will provide a clear visibility of router congestion. In said situation where on-path routers were labelled as congested are now classified according to their actual fine-grain predicted utilization. The predicted output from proposed model will helps routing algorithm or congestion handlers to route data packets through least congestion node, thus suppressing the effect and spread of potential congestion hotspot in NoC. The scope of this work is limited to explore high performing SNN based fine-grain congestion prediction model for NoC architecture that will be able to integrate with routing algorithm/ congestion handlers to minimize impact of congestion and thus improves network throughput and latency.

III. Fine-grain Congestion Prediction Models

Routers are equipped with input buffers to provide space for incoming data packets. Switch contention happens inside a router which permits incoming data to queue at input channel buffers. Queuing of data at input buffers is the foremost reason for local congestion [8], [16], [19], [30], [31]. These buffers utilization values provide early indication of possible local congestion. Research suggests that Buffer Occupancy Level (BOL) is a key indicator in identifying potential local congestion. Congestion handled at local level will reduce impact of back-pressure across NoC. Bio-inspired spiking neural network provides low hardware and high performance solution for learning and classification of temporal patterns. This work proposed SNN based NoC congestion prediction model that utilize buffer occupancy information as a parameter for prediction of multi-level local congestion. Inspired from previous work [13], this work considered two prediction models 1) Network model and 2) Router model to explore optimal low-cost model for multi-layer congestion prediction. Performance of both models are evaluated based on multi-level fine-grain prediction accuracy and additional resource costs.

A. Neural Model

SNNs are inspired from biological neurons to encode information in the form of spikes. Neurons in SNNs are connected though weighted synapses. Depending on level of abstraction, number of neural models and spiking learning algorithms are proposed for learning of temporal SNNs. This work proposed Leaky-integrate and Fire (LIF) model neurons with Spikeprop as a learning algorithm for training of NoC temporal utilization patterns. LIF model is an efficient spiking model and provides a true balance between neural behaviour and computational complexity [32]. In SNN, ith LIF neuron with membrane potential $u_i(t)$ at time t with leaking potential in the absence of an input current is described by

$$\tau_m \frac{d(u_i)}{dt} = -u_i(t) + RI_i^{syn}(t), \quad (1)$$

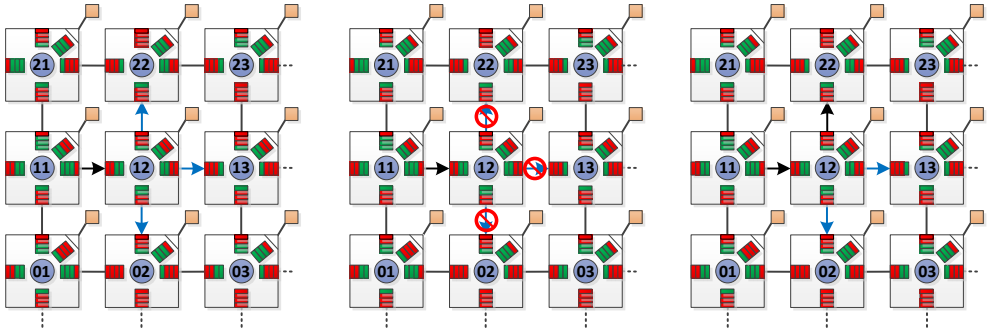
where τ_m describes the membrane time constant of the neuron, R is the membrane resistance and $I_i^{syn}(t)$ is synaptic current of ith neuron.

Spikeprop is SNN counterpart of the back-propagation in ANN which learns to updates the cost function by minimizing the (error) squared difference E between actual firing times t_j^a at output neuron j and desired firing time t_j^d at output neuron j by

$$E = \frac{1}{2} \sum_{j \in J} (t_j^a - t_j^d)^2. \quad (2)$$

B. Proposed Prediction models

The proposed SNN based fine-grain utilization prediction model connects directly with the NoC architecture and forwards



(a) (b) (c)

Fig 2. (a) available routing paths, (b) path prediction using bi-level congestion definition (c) path prediction using fine-grain congestion definition

data towards the routing algorithm to make routing decisions. The ultimate goal of the proposed models is to compare and contrast prediction performance and hardware area. This work considered two prediction models based on the level of abstraction: (1) router-model and (2) network-model. In router model, each router is connected with its individual SNN and reads BOL data from each input channel to predict fine-grain router congestion levels (shown in Fig 3(a)). The BOL can be accumulated as a unified value to show the Router Occupancy Level (ROL). The network model comes with a singular SNN for whole NoC architecture and reads ROL of each router in the NoC to predict a congestion level for each router (as shown in Fig 3(b)).The proposed models are designed to predict fine-grain utilization levels of each router 30 clocks ahead and informs the routing algorithm to take appropriate action to avoid potential congestion.

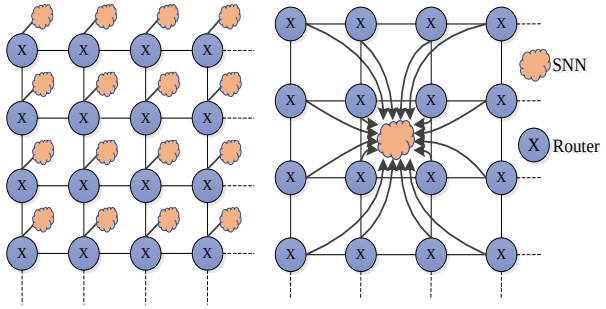


Fig 3. Proposed predicition models (a). Router model and (b) network model

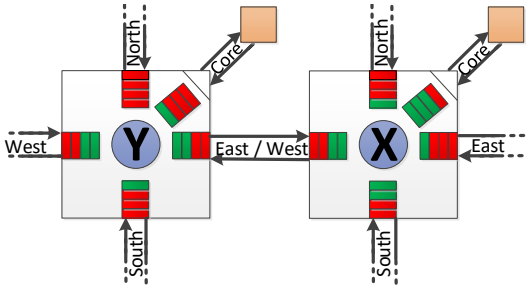


Fig. 4. Buffer Utilization model with 4-buffer slots for each input (Green are free slots; Red are occupied slots)

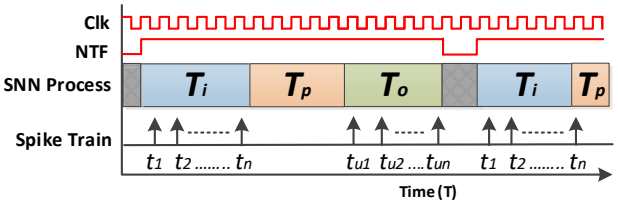


Fig. 5. Timing diagram of SNN.

C. *Neural Encoding*

The NoC architecture transmits packetized data between source and destination nodes through communication channels. Routers at every node establish these channel towards neighboring nodes. Depending on location in NoC (inner or corner), router can link through 3-5 neighboring channels. Each channel exhibit input buffers to accommodate incoming data packets towards node. The congestion level of the router depends on the total input buffer occupancy at connected channels. Fig 4 shows buffering information of the router-X and the router-Y in NoC network. It is evident from Fig 4 that both routers are connected to the neighboring routers and the processing core through 5 channels (north, west, south, east and core). The input buffer occupancy values for the router-X and the router-Y are $(3, 2, 2, 3, 1)$ and $(4, 2, 3, 2, 3)$ respectively. The ROL values of the router-X and the router-Y are 11 and 14 respectively. These buffer utilization values (BOL and ROL) are temporal in nature and can be fed directly into LIF-based SNN network for encoding of multi-level prediction. Fig.5 shows timing diagram for proposed SNN model where "T_i" is input spike time carrying BOL/ROL values $(t_1, t_2 \ldots t_n)$ as input spike train, "T_p" is neural processing time to generate output (predicted utilization values as spike ($t_{u1}, t_{u2} \ldots t_{un}$)) in time "T_o". For any input buffer pattern, the proposed SNN models will predict fine-grain congestion with 30 clocks in advance. This will provide enough time to routing mechanism to make decision for routing data through least congested path.

Referring to Fig.2(c), our proposed fine-grain models predicts actual router utilization and provides more visibility by determining future router utilization values based on current utilization status to the routing algorithm /congestion handlers to forwards routing data from R12 towards destination node. The proposed router and network models read real-time BOL and ROL values of R22, R13 and R02 to predict multi-level fine-grain utilization values (for reference let us assume 11, 14 and 12) for R22, R13 and R02 respectively. These fine-grain multi-level congestion values are shared with neighboring nodes and used by congestion handler/adaptive routing algorithms of to avoid potential hazard by routing data towards potentially least congested node R22 (as shown in Fig 2(b)).

Forwarding of data towards R22 using fine-grain prediction model will not only prevent NoCs from the effect of backpressure but also help to reduce delays and NoC latency. The scope of this paper is limited to exploring the feasibility of fine-grain multi-level congestion prediction with high accuracy and low hardware area overheads.

IV. Experimental Setup

This section explains the experimental setup established to implement the proposed SNN based fine-grain level congestion prediction models. The prediction models are tested on traced based synthetic and real-time multimedia application data for performance evaluation. These applications are mapped on Noxim, a cycle accurate NoC simulator to generate utilization patterns for each input buffer of NoC router [33]. In the router model, temporal utilization patterns are generated from Noxim are fed into SNN to predict local fine-grain level congestion for each network node. In the network prediction model these temporal buffer utilization packets are passed through accumulator to generate unified temporal ROL values for each NoC router before forwarding it to network-model SNN.

A. Traffic Scenarios

We considered traced-based synthetic and real-time Multi-media applications to analyze accuracy of fine-grain prediction models. All these applications are mapped on Noxim simulator to generate traffic patterns.

Noxim comes with preinstalled synthetic applications which are readily available for performance evaluation i.e. transpose1, transpose2 Butterfly and Shuffle traffic patterns[33]. Two multimedia applications, MMS and MPEG-4 applications are anticipated as benchmarks for real-time traffic scenarios in NoC architecture. MMS is a heterogeneous architecture comprising up of 40 tasks to be mapped on defined NoC Intellectual Property (IP) cores [34]. MPEG-4 is a video decoder distributed traffic traces mapped across 12 IPs which communicates through shared SDRAM [35].

B. Simulation Setup

The simulation environment considered a 4x4 NoC with standard 2-D XY routing algorithm for transmission of data in a mesh-based NoC. NoC throughput depends on the Packet Injection Rate (PIR). PIR is increased (PIR=0.5) to saturate network traffic to generate network congestion. Synthetic and real-time multimedia applications are mapped on Noxim and simulations are run for 2,000 clocks cycles to generate BOL and ROL of each router. Extracted data is then split into 60:40 for training and testing of the SNN.

Work proposed LIF based SNNs prediction model with SpikeProp as learning algorithm. Both prediction model SNNs are designed and simulated in MATLAB for training and testing. In router model, every router has its own SNN where input layer neurons are connected directly to input channel buffers. Size of SNN in router level depends on location of router in NoC architecture and requires 3-5 input neurons. Router level SNN comes with fully connected 3-layer [(3-5) x 15 x 1] sized SNN with (3-5), 15 and 1 neurons in input, hidden and output layer respectively are used for prediction of fine-grain router congestion. Contrary to the router model, network model requires single SNN to predict fine-grain congestion level for whole network node. A (16x30x16) network level neural predictor with 16 neurons at the input layer, 30 in the hidden and 16 in the output layers for a 4x4 NoC. Each input is connected with a NoC router to read ROL. The output layer produces a predicted congestion level for the respective router with 30 clocks in advance.

V. Results and Analysis

A. Performance Parameter

To analyses performance of proposed SNN based congestion prediction models, fully-connected LIF-spikeprop is developed in MATLAB. During training sessions, 60% of the utilization data is simulated by the SNN to achieve a learning accuracy of less than 5% Mean Square Error (MSE). Once training accuracy is achieved, the trained SNN models are then fed with unseen (40% of the dataset) to predict fine-grain congestion 30 clocks in advance. A prediction accuracy (P_a) parameter is used for the evaluation of multi-level fine-grain prediction results of each traffic application generated by SNN. It is defined by

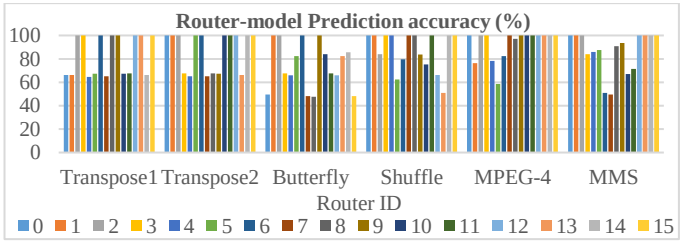
$$P_a = \frac{(\sum TP + \sum TN)}{\sum (P + N)} \quad (3)$$

where congestion patterns are termed as positive (P) and non-congestion patterns are labelled as negative (N). TP and TN defines correct prediction of patterns (P) and (N) respectively.

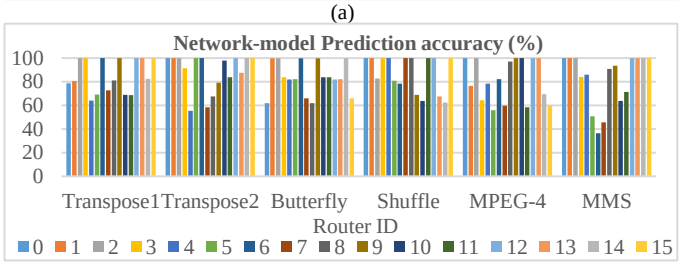
Overall performance of the fine-grain prediction models are evaluated on average NoC prediction performance and estimated hardware area overhead.

B. Simulation Analysis

Results generated by the SNNs are analyzed by the prediction accuracy of each router as well as an average NoC prediction accuracy under varying synthetic and real-time traffic patterns. Fig 6 shows prediction accuracy of each node using router-model and network-model. Fig 7 shows the



(a)



(b)

Fig 6. Prediction accuracy for each router using (a) Router model and (b) Network model

average prediction accuracy of the router and network models under varying traffic conditions. For the synthetic traffic application, the router-model predicted fine-grain BOL with 74.69%-87.64% average accuracy as compared to 83.32% - 88.32% accuracy with the network-model. The router-model prediction accuracy in real-time traffic applications marked between 86.30%-93.27% as compared to 81-38%-82.65% in the network-model. In general, the router-model has shown prediction accuracy between 74.69% and 93.27% when compared with the network-model which predicted multi-level

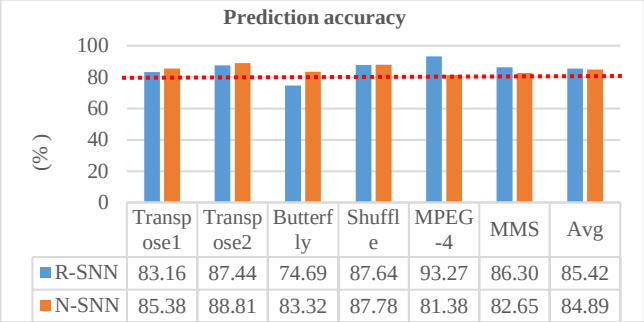


Fig 7. Average prediction accuracy of proposed prediction models

congestion accuracy between 81.38% and 88.81%.

The router model showed better average accuracy of 85.42% across all traffic applications as compared to 84.89% of network-model accuracy. Despite the slightly lower average prediction performance, the network-model showed more consistency in performance by keeping fine-grain congestion perdition accuracy above 80% across all traffic scenarios. Furthermore router-level shows the least prediction accuracy for multi-level congestion with 74.69% as compared to 81.23% in the network-model. Therefore, analysis suggests network-model equipped with a global SNN as a suitable solution for fine-grain congestion prediction in NoCs.

C. *Hardware analysis:*

The estimated hardware area overhead of the proposed fine-grain level congestion prediction models can be computed by using synaptic LIF neuron CMOS area calculated from [1], [36]. The router model requires an SNN for each node to predict congestion levels whereas the network router requires a single SNN to forecast congestion at each node. A typical CMOS based LIF neuron utilize $9 \times 10^{-6}\ mm^2$ and a synaptic neural interconnection costs $24 \times 10^{-8}\ mm^2$ CMOS area. Estimated hardware cost of the router and the network model predictors for a 4x4 NoC is shown in table 1.

TABLE 1 HARDWARE OVERHEAD

Simulator	Synaptic Area (mm^2)	Neural Area (mm^2)	Total Area (mm^2)	Predictor to Router Area Overhead (%)
Router model	1.92×10^{-3}	2.16×10^{-3}	4.08×10^{-3}	0.46
Network model	2.30×10^{-3}	5.58×10^{-4}	2.86×10^{-3}	0.32

It is illustrated that the router model requires $4.08 \times 10^{-3}\ mm^2$ CMOS area as compared to network model of $2.86 \times 10^{-3}\ mm^2$. When compared with the congestion aware adaptive router area [37], both models require fraction of area resources to add predictive capability to routing algorithm.

D. *Discussion*

Congestion prediction techniques requires complex algorithms to provide data transmission solution for scalable on-chip technology. Previous prediction models classify and labelled router congestion status as (1) congested (2) non-congested[13], [16], [20]. This classification pushes burden on the routing algorithms to decide for the next packet hop. Routing algorithms may require additional information in a condition where the next on-path routers are labelled as congested. The proposed prediction technique in this paper is multi-level congestion prediction which predicts the actual buffer occupancy for each NoC router with a 30 clock cycle look-ahead capability. The proposed SNN based models predicts congestion with 11% more accuracy and requires 9 times less hardware area when compared to the ANN based prediction model [16]. Previous SNN work based on bi-level congestion prediction has showed 92.83%-93.29% prediction accuracy for router and network congestion prediction models [13]. Although the proposed fine-grain prediction has reduced average prediction accuracy to 84.89%-85.42%, but it does provide in-depth router utilization prediction, which shifts anonymous routing decisions from the routing algorithm to allow sustained data distribution to enhance NoC throughput.

VI. CONCLUSION

This work proposed two SNN based fine-grain congestion prediction models for NoCs with a 30 clock cycle look ahead capability. The proposed models employed the LIF neural model with spikeprop as a learning algorithm to train the neural congestion prediction. The fine-grain prediction requires BOL for the router-model which reads input from each channel and ROL for the network-model which requires total occupancy for each node. Both models read inputs to predict multi-level fine-grain occupancy levels for each router. The output of the proposed model can be forwarded towards the congestion handlers i.e., adaptive routing algorithms to make routing decisions. The proposed models are trained and tested under synthetic and real-time traffic applications. Simulation results showed that the router-model predicted fine-grain congestion with an average accuracy of 85.42% as compare to 84.89% with the network-model. However, network model deliver consistency in prediction accuracy (more than 80% across all traffic scenarios) and requires 42% less hardware area than the network-model. Analysis suggests the network-model as more suitable approach for fine-grain congestion prediction in synthetic/real-time applications.

The proposed work is part of on-going research of exploring low-cost congestion prediction model to enhance NoC performance[13]. This work is limited to exploring suitable fine-grain prediction model which provides a trade-off between performance and hardware area. Future work includes analysis on Rentian traffic and the integration of the SNN based fine-grain congestion prediction model with an adaptive routing algorithm i.e. i.e., DyXY, DyAD, NoP and [4], [37] to improve NoC latency and throughput.

REFERENCES

[1] J. Harkin, F. Morgan, L. Mcdaid, S. Hall, B. Mcginley, and S. Cawley, "A Reconfigurable and Biologically Inspired Paradigm for Computation Using Network-On-Chip and Spiking Neural Networks," *Int. J. Reconfigurable Comput.*, vol. 2009, 2009.

[2] M. Amin, M. Shakir, A. Javed, M. Hassan, and S. A. Raza, "Low-cost fault tolerant methodology for real time MPSoC based embedded system," *Int. J. Reconfigurable Comput.*, vol. 2014, 2014.

[3] J. Liu, J. Harkin, Y. Li, and L. Maguire, "Online traffic-aware fault detection for networks-on-chip," *J. Parallel Distrib. Comput.*, vol. 74, no. 1, pp. 1984–1993, 2014.

[4] J. Liu, S. Member, J. Harkin, Y. Li, S. Member, and L. P. Maguire, "Fault-Tolerant Networks-on-Chip Routing With Coarse and Fine-Grained Look-Ahead," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 35, no. 2, pp. 260–273, 2016.

[5] U. Y. Ogras and R. Marculescu, "Prediction-based flow control for network-on-chip traffic," *Proc. DAC-44*, pp. 839–844, 2006.

[6] A. Charif, A. Coelho, and M. Nicolaidis, "MINI-ESPADA : A Low-Cost Fully Adaptive Routing Mechanism for Networks-on-Chips," *2017 18th IEEE Lat. Am. Test Symp.*, pp. 1–4, 2017.

[7] H. Tseng, R. Wu, W. Chang, Y. Lin, and D. Duh, "An Efficient Traffic-Based Routing Algorithm for 3D Networks-on-Chip," in *Int'l Conf. Embedded Systems, Cyber-physical Systems, & Applications (ESCS'16)*, 2016, pp. 73–79.

[8] M. Tang, "Analysis on Local Congestion of Network-on-Chip," in *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*, 2013, no. Iccsee, pp. 2863–2866.

[9] J. R. De Oliveira Neto, J. P. C. Cajueiro, and J. Ranhel, "Neural encoding and spike generation for Spiking Neural Networks implemented in FPGA," *25th Int. Conf. Electron. Commun. Comput. CONIELECOMP 2015*, pp. 55–61, 2015.

[10] J. Latif, C. Xiao, A. Imran, and S. Tu, "Medical Imaging using Machine Learning and Deep Learning Algorithms: A Review," *2019 2nd Int. Conf. Comput. Math. Eng. Technol.*, pp. 1–5, 2019.

[11] J. Harkin, F. Morgan, S. Hall, P. Dudek, T. Dowrick, and L. Mcdaid, "RECONFIGURABLE PLATFORMS AND THE CHALLENGES FOR LARGE-SCALE IMPLEMENTATIONS OF SPIKING NEURAL NETWORKS," in *International Conference on Field Programmable Logic and Applications ,Heidelberg, Germany*, 2008, pp. 483–486.

[12] W. Maass, "On the relevance of time in neural computation and learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1997.

[13] A. Javed, J. Harkin, L. Mcdaid, and J. Liu, "Exploring Spiking Neural Networks for Prediction of Traffic Congestion in Networks-on-Chip," in *IEEE International Symposium on Circuits and Systems (ISCAS), Seville Spain 2020 (accepted)*, 2020.

[14] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training Deep Spiking Neural Networks using Backpropagation," *CoRR*, vol. abs/1, pp. 1–10.

[15] B. Han, A. Sengupta, and K. Roy, "On the Energy Benefits of Spiking Deep Neural Networks : A Case Study," in *International Joint Conference on Neural Networks (IJCNN)*, 2016, no. 1, pp. 971–976.

[16] E. Kakoulli, V. Soteriou, and T. Theocharides, "Intelligent hotspot prediction for network-on-chip-based multicore systems," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 31, no. 3, pp. 418–431, 2012.

[17] M. Rezaei-ravari, "Low Latency Path Prediction Mechanism in 2D - NoC," *Electr. Eng. (ICEE), Iran. Conf.*, pp. 1565–1570, 2018.

[18] A. Javed, J. Harkin, L. Mcdaid, and J. Liu, "Minimising Impact of Local Congestion in Networks- on-Chip Performance by Predicting Buffer Utilisation," in *31st. Irish Signals and Systems Conference (ISSC), Letterkenny, Ireland*.

[19] E. J. Chang, H. K. Hsin, S. Y. Lin, and A. Y. Wu, "Path-congestion-aware adaptive routing with a contention prediction scheme for network-on-chip systems," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 33, no. 1, pp. 113–126, 2014.

[20] H. Cai, Y. Yang, F. Qu, J. Wu, and B. Wang, "Congestion Prediction Algorithm for Network on Chip," vol. 11, no. 12, pp. 7392–7398, 2013.

[21] W. J. Dally, "Virtual-channel flow control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 3, no. 2, pp. 194–205.

[22] W. Zhang, L. Hou, J. Wang, S. Geng, and W. Wu, "Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip," in *2009 WRI Global Congress on Intelligent Systems*, 2009, vol. 3, pp. 329–333.

[23] Ming Li, Qing-An Zeng, and Wen-Ben Jone, "DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip," *2006 43rd ACM/IEEE Des. Autom. Conf.*, pp. 849–852, 2006.

[24] J. Hu and R. Marculescu, "DYAD - Smart Routing for Networks-on-Chip," in *DAC 2004, June 7-1 1.2004, San Diego, California, USA*, pp. 260–263.

[25] P. Huang and W. Hwang, "An Adaptive Congestion-Aware Routing Algorithm for Mesh Network- on-Chip Platform."

[26] G. Ascia, V. Catania, M. Palesi, I. C. Society, D. Patti, and I. C. Society, "Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip," *IEEE Trans. Comput.*, vol. 57, no. 6, pp. 809–820, 2008.

[27] E. Carvalho, N. Calazans, and F. Moraes, "Congestion-aware task mapping in NoC-based MPSoCs with dynamic workload," *Proc. - IEEE Comput. Soc. Annu. Symp. VLSI Emerg. VLSI Technol. Archit.*, no. April, pp. 459–460, 2007.

[28] Y. S. C. Huang, K. C. K. Chou, and C. T. King, "Application-driven end-to-end traffic predictions for low power NoC design," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 21, no. 2, pp. 229–238, 2013.

[29] T. Chen, W. Fu, B. Xie, and C. Wang, "Packet triggered prediction based task migration for network-on-chip," *Microprocess. Microsyst.*, 2014.

[30] S. T. Atik, M. M. Imran, J. N. Mahi, J. A. Jeba, Z. I. Chowdhury, and M. S. Kaiser, "An Adaptive Routing Algorithm for on-chip 2D Mesh Network with an Efficient Buffer Allocation Scheme," in *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*, 2018, pp. 1–4.

[31] E. Nilsson, M. Millberg, J. Oberg, and R. Robin, "Load distribution with the Proximity Congestion Awareness in a Network on Chip," in *Proceedings of the Design,Automation and Test in Europe Conference and Exhibition (DATE'03)*, 2003, pp. 11126–11127.

[32] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, "Method for training a spiking neuron to associate input-output spike trains," in *IFIP Advances in Information and Communication Technology*, 2011.

[33] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Noxim : An Open , Extensible and Cycle-accurate Network on Chip Simulator," *2015 IEEE 26th Int. Conf. Appl. Syst. Archit. Process.*, pp. 162–163, 2015.

[34] S. Khan, S. Anjum, U. A. L. I. Gulzari, M. K. Afzal, T. Umer, and F. Ishmanov, "An Efficient Algorithm for Mapping Real Time Embedded Applications on NoC Architecture," *IEEE Access*, vol. 6, pp. 16324–16335, 2018.

[35] N. Architectures, K. Srinivasan, and K. S. Chatha, "A Technique for Low Energy Mapping and Routing in," in *ISLPED '05. Proceedings of the 2005 International Symposium on Low Power Electronics and Design, August 8-10,2005, San Diego, California, USA*, 2005, pp. 387–392.

[36] J. Liu, J. Harkin, M. Mcelholm, and L. Mcdaid, "Case Study : Bio-inspired Self-adaptive Strategy for Spike-based PID Controller," *2015 IEEE Int. Symp. Circuits Syst.*, pp. 2700–2703, 2015.

[37] S. Carrillo *et al.*, "Advancing interconnect density for spiking neural network hardware implementations using traffic-aware adaptive network-on-chip routers," *Neural Networks*, vol. 33, pp. 42–57, 2012.